2023013168 李震康

第一章

一. 名词解释

1. 操作系统是计算机中管理硬件和软件资源的系统软件

2. 实时操作系统是能优先处理紧急任务的操作系统

3. 互斥共享是某一个资源一个时间段只允许一个进程访问该资源

三. 填空

1. 硬件 软件

3. 硬件资源, 软件资源

5. 共享, 异步

第二章

一. 名词解释

3. 临界区是访问共享资源的那段代码

4. 进程同步是多道程序下, 对并发执行进程协调, 使按照一定顺序访问共享资源.

三. 填空

1. 动态性 异步性 结构性

5. 动态 静态

6. 进程互斥

8. 共享内存 消息传递 共享文件

9. 就绪 运行 阻塞

五. 问答

1. 前驱关系

$S_1 \to S_2$ $S_1 \to S_3$ $S_3 \to S_5$

$S_2 \to S_4$ $S_2 \to S_5$ $S_3 \to S_6$

$S_4 \to S_7$ $S_5 \to S_7$ $S_1 \to S_7$

```
main(){
    semaphore a,b,c,d,e,f,g = 0,0,0,0,0,0,0
    cobegin
    {S1; signal(a); signal(b);}        {wait(d); S5; signal(
    {wait(a); S2; signal(a); signal(c);}  {wait(e); wait(f); wa
    {wait(b); S3; signal(e);}                                  S6;}
    {wait(c); S4; signal(f);}
    coend;
```

三.填空

2.提交

6.预防死

五.问答

1. FCFS

SJF (非抢)

SJF (抢)

HRR

RRR

FBQ

FBQ (立即)

3.

```
cobegin
semaphore  empty=5, orange=0, apple=0, mutex=1;

Father(){
  while(1){
    wait(empty);
    wait(mutex);
    将水果放入盘子中;
    signal(mutex);
    if(是橘子)
        signal(orange);
    else signal(apple);
  }
}

Daughter(){
  while(1){
    wait(apple);
    wait(mutex);
    从盘中取一个苹果;
    signal(mutex);
    signal(empty);
    享用
  }
}

Son(){
  while(1){
    wait(orange);
    wait(mutex);
    从盘中取一个橘子;
    signal(mutex);
    signal(empty);
    享用
  }
}
```

4.

```
semaphore empty1=1, full1=0, empty2=1, full2=0

PA(){
  while(1){
    从磁盘读一个记录;
    wait(empty1);
    记录放到缓冲区1中;
    signal(full1);
  }
}

PB(){
  while(1){
    wait(full1);
    从缓冲区取一个记录;
    signal(empty1);
    wait(empty2);
    记录复制到缓冲区2;
    signal(full2);
  }
}

PC(){
  while(1){
    wait(full2);
    从缓冲区2中取一个记录;
    signal(empty2);
    将读取出的记录输打印
  }
}

main(){
  cobegin PA();
          PB();
          PC();
}
```

5.
```
int count = 0;
semaphore mutex = 1, empty = 1, full = 0;
semaphore payment = 0, receipt = 0;
guest() {
    wait (mutex);
    if (count >= N) {
        signal (mutex);
        离开;
    }
    else { count ++;
        signal (mutex);
        寻找座位;
        wait (empty);
        到理发椅就座;
        wait (mutex);
        count --;
        signal (mutex);
        signal (full);
        理发;
        付费;
        signal (payment);
        wait (receipt);
        signal (empty);
        离开;
    }
}
```

```
Barber() {
    while(1) {
        wait (full);
        理发;
        wait (payment);
        收费;
        signal (receipt);
    }
}
```

第二章

一. 名词解释

2. 对处理机进行分配

3. 指从作业被提交系统开始到作业完成为止这段时间间隔

4. 又双方都理对方释放自己所需资源，但都不能获取而无法释放自己资源.

三.填空

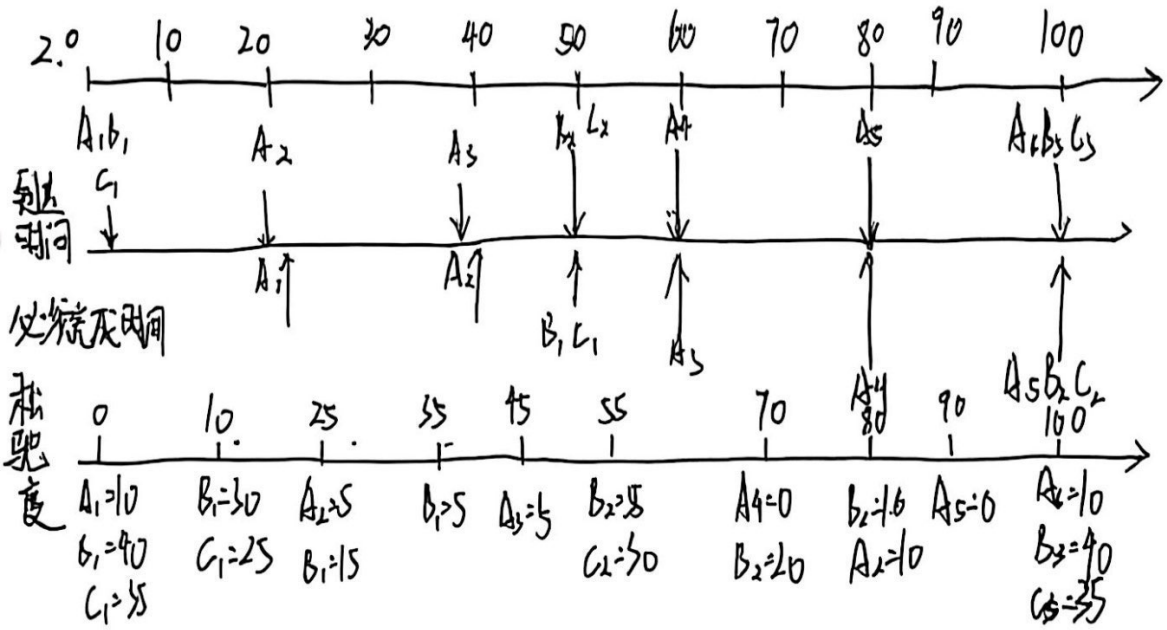
2.提交 后备完成

6.预防死锁 避免死锁 检测死锁 解除死锁

五.问答

		A	B	C	D	E	平均
1. FCFS	完成	3	9	13	18	20	
	周转	3	7	9	12	22	8.6
	带权	1	1.17	2.25	2.40	6	2.56
SJF (非抢)	完	3	9	15	20	11	
	周	3	7	11	14	3	7.6
	带	1	1.17	2.75	2.8	15	1.84
SJF (抢)	完	3	15	8	20	10	
	周	3	13	4	14	2	7.2
	带	1	2.16	1	2.8	1	1.59
HRRN	完	3	9	13	20	15	
	周	3	7	9	14	7	7.2
	带	1	1.17	2.25	2.8	3.50	1.59
RR(q=1)	完	4	18	17	20	15	
	周	4	16	13	14	7	8
	带	1.33	2.67	3.25	2.8	3.50	2.14
FB($q=2^{i-1}$)	完	3	17	18	20	14	
	周	3	15	14	14	6	10.4
	带	1	2.5	3.5	2.8	3	2.56
FB($q=2^{i-1}$) (立即抢占)	完	4	18	15	20	16	
	周	4	16	11	14	8	10.6
	带	1.33	2.67	2.75	2.8	4	2.87

2.



作业执行

3. 由 Allocation = Max - Need

Process	Allocation
P_0	0 0 3 2
P_1	1 0 0 0
P_2	1 3 5 4
P_3	0 3 3 2
P_4	0 0 1 4

资源情况 / 进程	Work	Need	Allocation	Work + Allocation	Finish
	A B C D	A B C D	A B C D	A B C D	
P_0	1 6 2 2	0 0 1 2	0 0 3 2	1 6 5 4	True
P_3	1 6 5 4	0 6 5 2	0 3 3 2	1 9 8 6	True
P_4	1 9 8 6	0 6 5 6	0 0 1 4	1 9 9 10	True
P_1	1 9 9 10	1 7 5 0	1 0 0 0	2 9 9 10	True
P_2	2 9 9 10	2 3 5 6	1 3 5 4	3 12 14 14	True

(2) P_2发出请求向量 Request (1,2,2,2)后，系统按银行家算法进行

检查：① Request (1,2,2,2) ≤ Need (2,3,5,6)

② Request (1,2,2,2) ≤ Available (1,6,2,2)

③ 系统先假定可为P_2分配资源，并修改 Available、Allocation，和 Need 向量 Available = (0,4,0,0)

Allocation = (2,5,7,6)

Need = (1,1,3,4)

④ 进行安全性检查，此时对所有进程，条件 Need ≤ Available (0,4,0,0) 都不成立，

∴进入不安全状态

∴不分配资源

也可，∵此时上述进程并没有申请到所需资源，并因得不到资源而进入阻塞状态，只有当上述进程提出新请求导致所有未执行完的多个进程都因得不到资源而得不到资源而阻塞并形成循环等待时，系统才进入死锁